

WHAT IS CLAIMED IS:

1. A method of debugging code containing a user-specified breakpoint located within a region of the code, the method comprising:
 - executing the code;
 - determining whether the execution of the code exits the region of the code without firing the user-specified breakpoint; and
 - if so, halting the execution of the code.
2. The method of claim 1, wherein the user-specified breakpoint is a conditional breakpoint having an associated condition and execution exits the region without firing the user-specified breakpoint because the associated condition is not satisfied.
3. The method of claim 1, further comprising:
 - encountering the user-specified breakpoint;
 - suspending the execution of the code at the user-specified breakpoint; and
 - in response to a user-specified run-to command received while execution of the code is suspended, executing the code until reaching an exit point of the region is reached.
4. The method of claim 1, wherein the user-specified breakpoint is a non-conditional breakpoint and execution exits the region without firing the user-specified breakpoint because the user-specified breakpoint is not encountered.
5. The method of claim 1, further comprising, prior to determining:
 - setting an internal safety net entry breakpoint in the code relative to an entry point of the region; and
 - setting a safety net exit breakpoint in the code relative to an exit point of the region.
6. The method of claim 5, wherein setting the internal safety net entry breakpoint and setting the safety net exit breakpoint are performed automatically in response to a user selection of the region.

7. The method of claim 5, wherein setting the safety net exit point is performed automatically in response to encountering the internal safety net entry breakpoint.

8. The method of claim 5, wherein the entry point and the exit point are determined by a compiler.

9. The method of claim 1, wherein the region of the code is defined by an entry point and an exit point and wherein halting comprises encountering a safety net breakpoint located in the code relative to the exit point.

10. The method of claim 9, further comprising determining, by a compiler, the entry point and the exit point prior to executing the code.

11. A computer readable medium containing a debug program which, when executed, performs an operation for debugging code containing a user-specified breakpoint located within a predetermined region of the code, the operation comprising:

determining whether the execution of the code exits the region of the code without firing the user-specified breakpoint; and
if so, halting the execution of the code.

12. The computer readable medium of claim 11, wherein the user-specified breakpoint is a conditional breakpoint having an associated condition and execution exits the region without firing the user-specified breakpoint because the associated condition is not satisfied.

13. The computer readable medium of claim 11, wherein the user-specified breakpoint is a non-conditional breakpoint and execution exits the region without firing the user-specified breakpoint because the user-specified breakpoint is not encountered.

14. The computer readable medium of claim 11, wherein the operation further comprises, prior to halting the execution of the code:

setting an internal safety net entry breakpoint in the code relative to an entry point of the region; and

setting a safety net exit breakpoint in the code relative to an exit point of the region.

15. The computer readable medium of claim 14, wherein setting the internal safety net entry breakpoint and setting the safety net exit breakpoint are performed automatically in response to a user selection of the region.

16. The computer readable medium of claim 14, wherein setting the safety net exit point is performed automatically in response to encountering the internal safety net entry breakpoint.

17. The computer readable medium of claim 14, wherein the entry point and the exit point are determined by a compiler.

18. The computer readable medium of claim 11, wherein the region of the code is defined by an entry point and an exit point and wherein halting comprises encountering a safety net breakpoint located in the code relative to the exit point.

19. The computer readable medium of claim 18, wherein the entry point and the exit point are determined by a compiler.

20. A computer system, comprising:

a memory containing at least a debug program and code containing a user-specified breakpoint located within a predetermined region of the code; and

a processor which, when executing content of the memory, is configured to perform an operation comprising:

determining whether the execution of the code exits the region of the code without firing the user-specified breakpoint; and

if so, halting the execution of the code.

21. The computer system of claim 20, wherein the user-specified breakpoint is a conditional breakpoint having an associated condition and execution exits the region without firing the user-specified breakpoint because the associated condition is not satisfied.

22. The computer system of claim 20, wherein the user-specified breakpoint is a non-conditional breakpoint and execution exits the region without firing the user-specified breakpoint because the user-specified breakpoint is not encountered.

23. The computer system of claim 20, wherein the operation further comprises, prior to halting the execution of the code:

setting an internal safety net entry breakpoint in the code relative to an entry point of the region; and

setting a safety net exit breakpoint in the code relative to an exit point of the region.

24. The computer system of claim 23, wherein setting the internal safety net entry breakpoint and setting the safety net exit breakpoint are performed automatically in response to a user selection of the region.

25. The computer system of claim 23, wherein setting the safety net exit point is performed automatically in response to encountering the internal safety net entry breakpoint.

26. The computer system of claim 23, wherein the entry point and the exit point are determined by a compiler.

27. The computer system of claim 20, wherein the region of the code is defined by an entry point and an exit point and wherein halting comprises encountering a safety net breakpoint located in the code relative to the exit point.

28. The computer system of claim 27, wherein the entry point and the exit point are determined by a compiler.

[illegible]